UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/679,635	10/06/2003	James A. Sievert	5575 (USYS.061PA)	1805

27516 7590 03/07/2007
UNISYS CORPORATION
MS 4773
PO BOX 64942
ST. PAUL, MN 55164-0942

EXAMINER	
ZHEN, LI B	
ART UNIT	**PAPER NUMBER**
2194	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/07/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

PTOL-90A (Rev. 10/06)

Office Action Summary	**Application No.** 10/679,635	**Applicant(s)** SIEVERT, JAMES A.
	Examiner Li B. Zhen	**Art Unit** 2194

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 December 2006*.
2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.
3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24* is/are pending in the application.
  4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5)☐ Claim(s) _____ is/are allowed.
6)☒ Claim(s) *1-7,11-18 and 20-24* is/are rejected.
7)☒ Claim(s) *8-10 and 19* is/are objected to.
8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.
10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.
  Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
  Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
  a)☐ All b)☐ Some * c)☐ None of:
    1.☐ Certified copies of the priority documents have been received.
    2.☐ Certified copies of the priority documents have been received in Application No. _____.
    3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
  * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
  Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413)
  Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1 – 24 are pending in the application.

Allowable Subject Matter

2. Claims 8 – 10 and 19 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Response to Arguments

3. Applicant's arguments filed 12/11/2006 have been fully considered but they are not persuasive. In response to the Non-Final Office Action dated 08/09/2006, applicant argues:

(1) The cited teachings of Mein do not suggest that the objects in the object model are in any apparent manner traversed [p. 10];

(2) There is no apparent suggestion or indication that Smith-1's dispatcher creates any event queue in response to a keyword in an HTTP request [pp. 10 – 11];

(3) The events in Smith-1 are not returned to the client as claimed [p. 11];

(4) There is no evidence presented that Mein is in any way inefficient or inflexible in providing its communication interfaces [p. 11];

(5) There is no apparent reasonable suggestion of any annotated property associated with the property name from the request. If the rejection is maintained, an explanation is requested as to the element of Ringseth thought to correspond to the

annotated property and the property name from the request [p. 12];

(6) The alleged motivation for combining Ringseth with Mein simply states the function of Ringseth without providing evidence of why one would be motivated to change Mein [p. 12];

(7) There is no citation of any portion of Smith-2's teachings that suggests that Smith-2 generates a unique identifier for the Value type class [p. 13];

(8) The alleged motivation for combining Smith-2 with Mein simply states the function of Smith-2 without providing actual evidence that would motivate one to modify Mein [p. 13].

In response to argument (1), examiner respectfully disagrees and notes that Mein discloses traversing the objects in the object model. Mein teaches identifying the Automation object with a progid that corresponds to an identifier contained in the HTTP POST message [col. 5, lines 39 – 52 and col. 6, line 58 - col. 7, line 10]. Mein discloses implemented the Automation objects as Component Object Model (COM) automation objects [col. 4, line 61 – col. 5, line 5]. Each COM object has a corresponding progid that is an alphanumeric representation of the unique GUID used to instantiate the automation object [col. 6, line 58 – col. 7, line 10]. Since Mein discloses a plurality of automation objects, the identification of the automation object based on the progid included in the HTTP POST message would include traversing the plurality of automation objects to identify the automation objects that corresponds to the progid. Applicant's specification also discloses that the HTTP handler parses the request, navigates the object model, instantiates a transient object with the

ExampleTransientObject property that included in the HTTP request, and returns the transient object to the client [p. 12, paragraph 0057]. Therefore, Mein teaches applicant's invention as claimed and disclosed.

As to argument (2), examiner disagrees and notes that the dispatcher of Smith-1 corresponds to the event queue because the dispatcher stores events and dispatches the events in an order [col. 5, lines 1 – 15] in response to a first keyword [col. 6, lines 45 – 54] obtained from an HTTP-form-processing request [col. 4, lines 25 – 56]. Smith-l discloses using the information in the SOAP message to create an event and firing the event to the SLEE engine using the dispatcher [col. 4, lines 47 – 56]. Therefore, Smith-1 discloses that the information in the SOAP message creates the event queue (dispatcher).

As to argument (3), examiner disagrees and notes that Smith-1 discloses that when processing of a request is complete, the Actor fires an event that contains a value object and the event containing the value object is sent to the client (initiator of the SOAP message) [col. 6, line 40 – 45 and col. 7, line 13 – 25].

In response to argument (4), the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, the motivation for combining the teachings of the prior art can be found in col. 1, lines 35 – 47 of Smith-1.

As to argument (5), examiner respectfully disagrees and notes that Ringseth teaches aborting parsing of an HTTP-form-processing request [message] in response to an annotated property [server class of errors] associated with a property name [processing of the message] obtained from the HTTP-form-processing request [Server class of errors indicate that the message could not be processed for reasons not directly attributable to the contents of the message itself, but rather attributable to the processing of the message; col. 13, lines 9 – 32]. Therefore, the combination of Mein and Ringseth teaches applicant's invention as recited in claims 4, 5, 15 and 16.

As to argument (6), examiner disagrees and notes that the motivation does not simply state the functions of Ringseth. One would have been motivated to combine the teachings of Mein and Ringseth because the combination provides the benefits of an mechanism for identifying the fault that allows the system stop processing of the message at runtime based on a parameter in the request message [col. 12, lines 17 – 23 and col. 20, lines 10 – 33 of Ringseth].

As to argument (7), examiner disagrees and notes that the currently recited claims do not recite generating a unique identifier for a value type class. Although the claims are interpreted in light of the specification, limitations from the specification (assuming that the specification discloses generating a unique identifier for a value type class) are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

As to argument (8), examiner disagrees and notes that the motivation does not simply state the functions of Ringseth. One would have been motivated to combine the teachings of Mein and Smith-2 because the combination provides the benefits of a discovery namespace that allow Web service consumers to locate available Web Services on a Web server [col. 7, lines 18 – 35 of Smith162] and enables rich management through late binding to providers [col. 940, lines 45 – 55 of Smith162].

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. **Claims 1,2,11-13 and 22-23 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,782,542 to Mein et al. [hereinafter Mein, cited in the previous office action].**

6. As to claim 1, Mein teaches a processor-implemented method for exposing an application object model [Automation objects are implemented as COM automation objects. COM is the Component Object Model; col. 4, line 61 - col. 5, line 5] to web-

based clients [client process 110 requires certain data from an Automation object, the process issues a method call; col. 5, lines 23 – 39], comprising:

parsing each hyper-text transport protocol (HTTP)-form-processing request [HTTP POST message; col. 5, lines 23 – 39] received from a client by a web server [SOAP stub 170 implements an HTTP parser that unpackages the multipart MIME packets into individual parameters for the method call; col. 7, lines 30 – 40];

traversing objects in the object model [progid is an alphanumeric representation of the unique GUID used to identify the Automation class to instantiate on the target machine; col. 6, line 58 - col. 7, line 10] in response each property name obtained from the HTTP-form-processing request [a method name 324 that identifies the object method to be invoked following the progid 22, and any [in] parameters 326 that are needed; col. 6, line 58 – col. 7, line 11];

wherein traversing includes comparing a property name [identifier contained in the header; col. 5, lines 39 – 52] from the request [HTTP POST message col. 5, lines 39 – 52] to a property name [progid; col. 6, line 58 - col. 7, line 10] of an object defined in the object model [instantiates the Automation object 180 identified in the header; col. 5, lines 39 – 52]; and

returning to the client [returns the results, or [out] parameters; col. 7, lines 45 – 55] a value of a selected property of an object in the object model in response to traversal of the object model [Variant and Automation "object" classes such as the ADO Recordset may be used as either [in], [out], or [in, out] parameters; col. 3, lines 35 – 52].

7. As to claim 12, Mein teaches a computing arrangement for exposing objects to web-based clients [a system for allowing a client process on a local client computer 10a, 10b, 10c, 10d, . . . to access and invoke Automation objects located on remote ISAPI-enabled Web servers 30a, 30b, 30c; col. 4, lines 30 – 40], comprising:

a web server arrangement that includes a data processing system configured with a web server [Web servers 30a, 30b, 30c; col. 4, lines 30 – 40];

an object model library [Automation classes; col. 6, line 58 – col. 7, line 10] hosted by the web server [Web server 160; col. 7, lines 15-32], the object model library containing the objects to be exposed [Automation objects are implemented as COM automation objects. COM is the Component Object Model; col. 4, line 61 - col. 5, line 5];

a web objects model [SOAP stub 170 is a server-side Applications Programming Interface (API) that interacts with the SOAP proxy 140; col. 7, lines 15 – 32] hosted by the web server [Web server 160; col. 7, lines 15-32], the web objects model configured to parse each hyper-text transport protocol (HTTP)-form-processing request [HTTP POST message; col. 5, lines 23 – 39] received from a client by a web server [SOAP stub 170 implements an HTTP parser that unpackages the multipart MIME packets into individual parameters for the method call; col. 7, lines 30 – 40], traverse objects in the object model [progid is an alphanumeric representation of the unique GUID used to identify the Automation class to instantiate on the target machine; col. 6, line 58 - col. 7, line 10] in response to property names obtained from the HTTP-form-processing request [a method name 324 that identifies the object method to be invoked following the progid 22, and any [in] parameters 326 that are needed; col. 6, line 58 – col. 7, line

11], wherein traversing includes comparing a property name [identifier contained in the header; col. 5, lines 39 – 52] from the request [HTTP POST message col. 5, lines 39 – 52] to a property name [progid; col. 6, line 58 - col. 7, line 10] of an object defined in the object model [instantiates the Automation object 180 identified in the header; col. 5, lines 39 – 52]; and

return to the client [returns the results, or [out] parameters; col. 7, lines 45 – 55] a selected property of an object in the object model in response to traversal of the object model [Variant and Automation "object" classes such as the ADO Recordset may be used as either [in], [out], or [in, out] parameters; col. 3, lines 35 – 52].

8. As to claim 23, Mein teaches an apparatus for exposing an application object model to web-based clients [a system for allowing a client process on a local client computer 10a, 10b, 10c, 10d, . . . to access and invoke Automation objects located on remote ISAPI-enabled Web servers 30a, 30b, 30c; col. 4, lines 30 – 40], comprising:

means [SOAP stub 170] for parsing each hyper-text transport protocol (HTTP)-form-processing request [HTTP POST message; col. 5, lines 23 – 39] received from a client by a web server [SOAP stub 170 implements an HTTP parser that unpackages the multipart MIME packets into individual parameters for the method call; col. 7, lines 30 – 40];

means for traversing objects in the object model [progid is an alphanumeric representation of the unique GUID used to identify the Automation class to instantiate on the target machine; col. 6, line 58 - col. 7, line 10 and col. 5, lines 38 – 52] in response to property names obtained from the HTTP-form-processing request [a method name 324 that identifies the object method to be invoked following the progid 22, and any [in] parameters 326 that are needed; col. 6, line 58 – col. 7, line 11] and for comparing a property name [identifier contained in the header; col. 5, lines 39 – 52] from the request [HTTP POST message col. 5, lines 39 – 52] to a property name [progid; col. 6, line 58 - col. 7, line 10] of an object defined in the object model [instantiates the Automation object 180 identified in the header; col. 5, lines 39 – 52]; and

means for returning to the client [returns the results, or [out] parameters; col. 7, lines 45 – 55] a selected property of an object in the object model in response to traversal of the object model [Variant and Automation "object" classes such as the ADO Recordset may be used as either [in], [out], or [in, out] parameters; col. 3, lines 35 – 52].

9. As to claim 2, Mein teaches initiating execution of a method of an object [method is invoked by the SOAP stub 170 using [in] parameters contained in the body of the data structure; col. 5, lines 38 – 52] in response to a method name obtained from an HTTP-form-processing request [method name field also identified in the header of the data structure indicates the method of the Automation object 180 to invoke; col. 5, lines 38 – 52].

10. As to claim 11, Mein teaches configuring the web server arrangement with an initial object name [name of the Automation object of interest; col. 3, lines 36 – 50]

associated with the a name of an application object model [Automation Object implemented as COM automation objects; col. 4, line 61 – col. 5, line 5]; and retrieving an object definition [Automation class] of the initial object name in response to the name of the application object model [progid is an alphanumeric representation of the unique GUID used to identify the Automation class; col. 6, line 58 - col. 7, line 10 and col. 5, lines 38 – 52] obtained from an HTTP-form-processing request [HTTP POST message also includes fields for holding data representing the PROGRAM ID (progid) 322 of the COM Automation object; col. 6, line 58 – col. 7, line 10].

11. As to claim 13, Mein teaches the web objects model is further configured to initiate execution of a method of an object [method is invoked by the SOAP stub 170 using [in] parameters contained in the body of the data structure; col. 5, lines 38 – 52] in response to a method name obtained from an HTTP-form-processing request [method name field also identified in the header of the data structure indicates the method of the Automation object 180 to invoke; col. 5, lines 38 – 52].

12. As to claim 22, Mein teaches an initial object name [name of the Automation object of interest; col. 3, lines 36 – 50] associated with the a name of an application object model [Automation Object implemented as COM automation objects; col. 4, line 61 – col. 5, line 5] in the web server; and wherein the web objects model is further configured to retrieve an object definition [Automation class] of the initial object name [progid is an alphanumeric representation of the unique GUID used to identify the

Automation class; col. 6, line 58 - col. 7, line 10 and col. 5, lines 38 – 52] in response to the name of the application object model obtained from an HTTP-form-processing request [HTTP POST message also includes fields for holding data representing the PROGRAM ID (progid) 322 of the COM Automation object; col. 6, line 58 – col. 7, line 10].

Claim Rejections - 35 USC § 103

13. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

14. Claims 3, 14 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mein in view of U.S. Patent No. 6,973,659 to Smith et al. [hereinafter Smith, cited in the previous office action].

15. As to claim 3, Mein does not disclose receiving events generated by application objects.

However, Smith teaches creating an event queue [dispatcher; it is noted that the dispatcher stores events and dispatches events in an order, col. 5, lines 1 – 15] in response to a first keyword [col. 6, lines 45 – 54] obtained from an HTTP-form-processing request [SOAP package (12) generates a SOAP message (that includes identification of the remote procedure) that is transmitted to the remote SOAP server;

col. 4, lines 25 – 40], registering, in response to a second keyword [a SOAP request stating Actor as the class name, myAction as the method name and myParameter as the parameter object; col. 6, lines 45 – 54] obtained from an HTTP-form-processing request, with the application objects to receive events generated by the application objects [dispatcher receives all events and passes them to service instances that have registered to receive the event; col. 5, lines 1 – 15], entering the events in the event queue [dispatcher receives all events; col. 5, lines 1 – 25], and returning events in the event queue to the client in response to a third keyword [col. 6, lines 45 – 54] obtained from an HTTP-form-processing request [SOAP message is returned to the client (39) and the client extracts the value from the SOAP message and returns it as the myAction method return value; col. 7, lines 1 – 32].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Smith and Mein because Smith's teachings provides efficient flexible mechanisms for multiple interfaces to communicate events with the call, thereby simplifying the service logic interfaces [col. 1, lines 35 – 47 of Smith].

16. As to claims 14 and 24, these claims are rejected for the same reasons as claim 3 above.

17. Claims 4, 5, 15 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mein in view of U.S. Patent No. 7,055,143 to Ringseth et al. [hereinafter Ringseth, cited in the previous office action].

18. As to claim 4, Mein does not disclose aborting parsing of an HTTP-form-processing request in response to an annotated property associated with a property name obtained from the HTTP-form-processing request.

However, Ringseth teaches a client sends a SOAP HTTP Request to a server and server responds to client via SOAP HTTP Response [col. 14, lines 15 – 22], parsing the request [col. 14, lines 32 – 43], and aborting parsing of an HTTP-form-processing request in response to an annotated property associated with a property name obtained from the HTTP-form-processing request [Server class of errors indicate that the message could not be processed for reasons not directly attributable to the contents of the message itself, but rather attributable to the processing of the message; col. 13, lines 9 – 32].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Ringseth and Mein because this provides an algorithmic mechanism for identifying the fault [col. 12, lines 17 – 23 of Ringseth] and this allows the system stop processing of the message at runtime based on a parameter in the SOAP request message [results in a runtime error if this header is not received; col. 20, lines 10 – 33 of Ringseth].

19. As to claim 5, Mein does not disclose converting a character string parameter in an HTTP-form-processing request to a value having a data type compatible with a property of a property name obtained from the HTTP-form-processing request and setting the property of the object to the value.

However, Ringseth teaches converting a character string parameter in an HTTP-form-processing request to a value having a data type [parses the XML of the message and marshals the parameters e.g., "inputString," into real C++ data types i.e., a string in this case; col. 14, lines 31 – 43] compatible with a property of a property name obtained [a string parameter as input i.e., a BSTR is a string type; col. 14, lines 8 – 16] from the HTTP-form-processing request and setting the property of the object to the value [calls the intended recipient of the "Hello World" function with the expected parameters e.g., "inputString"; col. 14, lines 31 – 43].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Ringseth and Mein because Ringseth's teachings provide this provides code to automatically generate SOAP messages from programming language constructs and generate of programming language constructs from SOAP messages and frees the developer to focus on the code to specify the description of the web service, and the code specific to their application [col. 14, line 55 – col. 15, line 13 of Ringseth].

20. As to claims 15 and 16, these are rejected for the same reasons as claims 4 and 5 respectively, see the rejections to claims 4 and 5 above.

21. Claims 6, 7, 17, 18, 20 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mein in view of U.S. Patent No. 7,017,162 et al. to Smith [hereinafter Smith162, cited in the previous office action].

22. As to claim 6, Mein discloses that the client has access to the global unique identifier of an Automation Class and the global unique identifier is assigned to a Automation class [col. 6, lines 58 – col. 7, line 10]. Mein does not specifically disclose for a reference-type property of an object found in traversing the object model, generating a unique identifier for the reference-type property, registering the unique identifier for the reference-type property in a collection, and returning a Uniform Resource Identifier (URI) that includes the unique identifier to the client. However, at some point in Mein's system prior to the client accessing the Automation Object, the global unique identifier would have to be created and registered with the object model in order for the client to access the Automation Class using a global unique identifier.

In addition, Smith162 teaches clients [clients 120; col. 4, lines 25 – 39] accessing object model services [col. 5, line 60 – col. 6, line 10] on a server [Web application 110 is composed of one or more software applications 130 that run atop a programming framework 132, which are executing on one or more servers 134; col. 4, lines 25 – 39] using a HTTP-form request [col. 3, lines 46 – 64], a reference-type property [reference type; col. 11, line 1 – col. 12, line 15] of an object found in traversing the object model [col. 7, lines 18 - 36], generating a unique identifier for the reference-type property [col.

300, lines 50 – 58], registering the unique identifier for the reference-type property in a collection [col. 882, lines 27 – 45], and returning a Uniform Resource Identifier (URI) that includes the unique identifier to the client [col. 526, lines 3 – 18 and col. 529, lines 6 – 26].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Smith162 and Mein because Smith162's teachings provides a discovery namespace containing classes that allow Web service consumers to locate available Web Services on a Web server [col. 7, lines 18 – 35 of Smith162] and enables rich management through late binding to providers [col. 940, lines 45 – 55 of Smith162].

23. As to claim 7, Mein as modified by Smith162 teaches the URI [col. 526, lines 3 – 18 and col. 529, lines 6 – 26 of Smith162] is a globally unique identifier (GUID) [col. 300, lines 50 – 58 of Smith162].

24. As to claims 17 and 18, these are rejected for the same reasons as claims 6 and 7 respectively, see the rejection to claims 6 and 7 above.

25. As to claim 20, Mein as modified by Smith162 teaches the web objects model is further configured to remove a short-lifetime type reference from the collection if a client does not submit an HPP-forms-processing request containing the URI for a selected period of time [common language runtime "garbage collection" mechanism reclaims

inaccessible (that is, "unreachable") memory allocated to an object. An object becomes unreachable if all references to it become invalid, for example, by setting those references to null; col. 524, lines 3 – 65 of Smith162].

26. As to claim 21, Mein as modified teaches the web objects model is further configured to maintain long-lifetime type reference in the collection until a referenced application object is removed [garbage collection; col. 298, lines 28 – 63 and col. 341, lines 43 – 50 of Smith162].

Conclusion

27. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

CONTACT INFORMATION

28. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Li B. Zhen whose telephone number is (571) 272-3768. The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Thomson can be reached on 571-272-3718. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Li B. Zhen
Examiner
Art Unit 2194

LBZ

MENGAL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100